State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups*

Ludovico Boratto and Salvatore Carta

Abstract Recommender systems are important tools that provide information items to users, by adapting to their characteristics and preferences. Usually items are recommended to individuals, but there are contexts in which people operate in *groups*. To support the recommendation process in social activities, group recommender systems were developed. Since different types of groups exist, group recommendation should adapt to them, managing *heterogeneity* of groups. This chapter will present a survey of the state-of-the-art in group recommendation, focusing on the type of group each system aims to. A new approach for group recommendation is also presented, able to adapt to technological constraints (e.g., bandwidth limitations), by automatically identifying groups of users with similar interests.

1 Introduction

Recommender systems aim to provide information items (web pages, books, movies, music, etc.) that are of potential interest to a user. To predict the items to suggest, the systems use different sources of data, like preferences or characteristics of users.

However, there are contexts and domains where classic recommender systems cannot be used, because people operate in *groups*. Here are some examples of such contexts:

Ludovico Boratto
Dipartimento di Matematica e Informatica, Università di Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy e-mail: boratto@sc.unica.it

Salvatore Carta
Dipartimento di Matematica e Informatica, Università di Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy e-mail: salvatore@unica.it

* This is a draft version of the paper to appear in Information Retrieval and Mining in Distributed Environments. Studies for Computational Intelligence, Springer Verlag, 2010.

- a system has to provide recommendations to an established group of people who share the same interests and do something together;
- recommendations are provided to an heterogeneous group of people who has a common, specific aim and shares the system on a particular occasion;
- a system tries to recommend items in an environment shared by people who don't have anything in common (e.g., background music in a room);
- when a limitation in the number of available recommendations to be provided is present, individuals with similar preferences have to be grouped.

To manage such cases, group recommendation was introduced. These systems aim to provide recommendations to groups, considering the preferences and the characteristics of more than a user. But what is a *group*? As we can see from the list above, there are at least four different notions of group:

1. **Established group**: a number of persons who explicitly choose to be a part of a group, because of shared, long-term *interests*;
2. **Occasional group**: a number of persons who do something occasionally together, like visiting a museum. Its members have a common *aim* in a particular moment;
3. **Random group**: a number of persons who share an environment in a particular moment, without explicit interests that link them;
4. **Automatically identifed group**: groups that are automatically detected considering the preferences of the users and/or the resources available.

Of course the way a group is formed affects the way it is modeled and how recommendations are predicted.

This chapter will present a survey of the state-of-the-art in group recommendation. A few years ago [29] presented a state-of-the-art survey too, dividing the group recommendation process into four subtasks and describing how each system handles each subtask. Here we will try to describe the existing approaches, focusing on the different notions of group and how the type of group affects the way the system works. Table 1 presents an overview of these systems. Moreover, we will present a new approach, proposed in [8], able to adapt to technological constraints and automatically detect groups of different granularities to fulfill the constraints.

The rest of the chapter is organized as follows: section 2 describes approaches that consider groups with an a priori known structure; section 3 considers systems that automatically identify groups and in 3.2 the new approach cited above is presented; in section 4 we will try to draw some conclusions.

System	**Domain of recommendation**	**Example of group**	**Type of group**
GRec_OC (Group Recommender for Online Communities) [31]	Books	Online communities that share preferences	1. Established group
Jukola [45]	Music	People attending a party	
PartyVote [53]	Music	People attending a party	
[47]	Movies	Interacting members that share opinions	
I-SPY [51, 50, 52, 49, 9, 22]	Web pages	Communities of like-minded users	
Glue [12]	Web pages	Online communities	
CAPS (Context Aware Proxy based System) [48]	Web pages	Colleagues that browse the web together	
[5]	Documents	Conference committees	
PolyLens [44]	Movies	People who want to see a movie together	2. Occasional group
[14]	Movies	People that share opinions	
[1]	Movies	People that share their *disagreement* with other members	
[18, 19]	Movies	People making decision for a group	
CATS (Collaborative Advisory Travel System) [36, 39, 40, 38, 37]	Travel vacation	Friends planning ski holidays	
INTRIGUE (INteractive TouRist Information GUidE) [3, 2]	Sightseeing destinations	People traveling together	
Travel Decision Forum [27, 26, 28]	Travel vacation	People planning a vacation together	
[33]	Travel vacation	People planning a vacation together	
e-Tourism [23]	Tourist tours	People traveling together	
Pocket RestaurantFinder [34]	Restaurants	People who want to dine together	
FIT (Family Interactive TV System) [25]	TV programs	Family members watching TV together	
[54]	TV programs	Family members watching TV together	
TV4M [56]	TV programs	People watching TV together	
Adaptive Radio [13]	Music	People who share an environment	3. Random group
In-Vehicle Multimedia Recommender [57]	Multimedia items	Passengers traveling together in a vehicle	
Flytrap [17]	Music	People in a public room	
MusicFX [35]	Music	Members of a fitness center	
Let's Browse [32]	Web pages	People that browse the web together	
GAIN (Group Adapted Interaction for News) [46, 11]	News items	People who share an environment	
[10]	Ontology concepts	People that share same interests	4. Automatically identified group
[8]	Movies	People with similar preferences	

Table 1 Overview of the existing group recommender systems

2 Group recommendation for groups with an a priori known structure

2.1 Systems that consider established groups

An *established group* is formed by people who share common interests for a long period of time. According to [44] established groups have the property to be *persistent* and users actively *join* the group.

As Table 1 shows, group recommender systems that aim to established groups are designed for domains of recommendation like:

- entertainment/cultural items (books, music and movies);
- documents (web pages and conferences documents).

2.1.1 Group recommender systems for entertainment/cultural items

GRec_OC (Group Recommender for Online Communities) [31] is a book recommender system for online communities (i.e., people with similar interests that share information). The system aims to improve satisfaction of individual users.

The approach works in two phases. Since the system aims to established groups, the first phase uses a classic Collaborative Filtering (CF) method to build a group profile, by merging the profiles of its members. Each group's nearest neighbors are found and a "candidate recommendation set" is formed by selecting the top-*n* items. To achieve satisfaction of each member, the second phase evaluates the relevance of the books in the candidate recommendation set for each member. Items not preferred by any member are eliminated and a list of books is recommended to the group.

Jukola [45] and *PartyVote* [53] are two systems able to provide music to an established social group of people attending a party/social event.

The type of group and the context in which the systems are used, make these systems work without any user profiles. In fact, in order to select the music to play, each user is allowed to express preferences (like the selection of a song, album, artist or genre) in a digital musical collection. The rest of the group votes for the available selections and a weight/percentage is associated to each song (i.e., the probability for the song to be played). The song with the highest vote is selected to be played.

The system proposed in [47] aims to produce personality aware group recommendations, i.e., recommendations that consider the personality of its members ("group personality composition") and how conflicts affect the recommendation process.

To measure the behaviors of people in conflicts, each user completes a test and a profile is built computing a measure called *Conflict Mode Weight (CMW)*. Recommendations are calculated using three classic recommendation algorithms, integrated with the CMWs of the group members.

2.1.2 Group recommender systems for documents

I-SPY [51, 50, 52, 49, 9, 22, 16] is a search engine that personalizes the results of a web search, using the preferences of a community of like-minded users.

When a user expresses interest in a search result by clicking on it, I-SPY populates a *hit matrix* that contains relations between the query and the results pages (each community populates its own matrix). Relations in the hit matrix are used to re-rank the search results to improve search accuracy.

Glue [12] is a collaborative retrieval algorithm that monitors the activity of a community of users in a search engine, in order to exploit implicit feedbacks.

A feedback is collected each time a user finds a relevant resource during a search in the system. The algorithm uses the feedback to dynamically strengthen associations between the resource indicated by the user and the keywords used in the search string. Retrieval is based on the feedbacks, so it's not just dependent on the resource's content, making it possible for the system to retrieve even non-textual resources and update its performances dynamically (i.e., the community of users decides which resources are described by which keywords).

CAPS (Context Aware Proxy based System) [48] is an agent that recommends pages and annotates links, based on their popularity among a user's colleagues and the user's profile. The system focuses on two aspects: page enhancement, with symbols that indicate its popularity, and search queries augmentation, with the addition of relevant links for a query. Since the system was designed to enhance the search activity of a user considering the experience of a user's colleagues, a CF approach and a zero-input interface (able to gather implicit information) were used.

The approach proposed in [5] was developed to help a group of conference committees selecting the most suitable items in a large set of candidates.

The approach is based on the *relative preference* of each reviewer, i.e., a rank of the preferred items, with no numeric score given to express the preferences. All the preferences ordering of the reviewers are aggregated through a variable neighborhood search algorithm improved by the authors for the recommendation purpose.

2.2 Systems that consider occasional groups with a particular aim

There are lots of contexts in which a group of people is not established but might be interested in getting together for a common aim. This is for example the case of people traveling together: they might not know each other, but they share interest for a common place. In such cases, a group recommender system could be useful, since it would be able to put together the preferences of an heterogeneous group, in order to achieve the common aim. As mentioned in Table 1, group recommender systems that work for occasional groups were developed for the following domains:

- movies;
- tourist destinations;
- TV programs;

Group recommender systems for TV programs consider occasional groups that get togeher for a specific aim (watch TV together) and randomly share an environment (approaches for random groups are described next). Since the approaches focus on the group's aim, this category of systems was placed in this subsection.

2.2.1 Group recommendation for movies

PolyLens [44] is a system built to produce recommendations to groups of users who want to see a movie. To produce recommendations for each user of the group a CF algorithm is used. The movies with the highest recommended rates are considered and a "least misery" strategy is used: the recommended rating for a group is the lowest predicted rating for a movie, to ensure that every member is satisfied.

The system proposed in [14] considers interactions among group members, assuming that in a group recommender system ratings are not given just by individuals, but also by subgroups. If a group G is composed of members u_1, u_2 and u_3, ratings might be given by both individuals and subgroups (e.g., $\{u_1, u_2\}$ and $\{u_1, u_3\}$).

The system learns the ratings of a group using a Genetic Algorithm (GA), that uses the ratings of both individuals and subgroups to learn how users interact. For example, if an item is rated by users u_1 and u_2 as 1 and 5 but as a whole they rate the item as 4, it is possible to derive that u_2 plays a more influential role in the group.

The group recommendation methodology used combines an item-based CF algorithm and the GA, to improve the quality of the system.

In [1] an approach to compute group recommendation that introduces *disagreement* between group members as an important aspect to efficiently compute group recommendations is presented. The authors introduce a *consensus function*, which combines *relevance* of the items for a user and *disagreement* between members. After the *consensus function* is built, an algorithm to compute group recommendation (based on the class of Threshold algorithms) is proposed.

The system proposed in [18, 19] presents a group recommendation approach based on Bayesian Networks (BN). The system was developed to help a group of people making decisions that involve the whole group (like seeing a movie) or in situations where individuals must make decisions for the group (like buying a company gift). The system was empirically tested in the movie recommendation domain.

To represent users and their preferences a BN is built. The authors assume that the composition of the groups is a priori known and model the group as a new node in the network that has the group members as parents. A collaborative recommender system is used to predict the votes of the group members. A posteriori probabilities are calculated to combine the predicted votes and build the group recommendation.

2.2.2 Group recommendation for tourist destinations

In [36, 39, 40, 38, 37] a group recommender system called *CATS (Collaborative Advisory Travel System)* is presented. Its aim is to help a group of friends plan and arrange ski holidays. To achieve the objective, users are positioned around a device called "DiamondTouch table-top" [20] and the interactions between them (since they physically share the device) help the development of the recommendations.

To produce the recommendations, the system collects *critiques*, which are feedbacks left by users while browsing the recommended destinations (e.g., a user might specify that he/she is looking for a cheaper hotel, by *critiquing* the price feature).

Interactions with the DiamondTouch device are used to build an individual personal model (IM) and a group user model (GUM). Individual recommendations are built using both the IM and the GUM to maximize satisfaction of the group, whereas group recommendations are based on the critiques contained in the GUM.

INTRIGUE (INteractive TouRist Information GUidE) [3, 2] is a system that recommends sightseeing destinations using the preferences of the group members.

Heterogeneity of a group is considered in several ways. Each group is subdivided into homogeneous subgroups of similar members that fit a stereotype (e.g., children). Recommendations are predicted for each subgroup and an overall preference is built considering some subgroups more influential (e.g., disabled people).

Travel Decision Forum [27, 26, 28] is a system that helps groups of people plan a vacation. Since the system aims to find an agreement between the members of a group, asynchronous communication is possible and, through a web interface, a member can view (and also copy) other members' preferences. Recommendations are made using a simple aggregation (the *median*) of the individual preferences.

In [33] a multiagent system in which agents work on behalf of a group of customers, in order to produce group recommendations, is presented. A formalism, named DCOP (Distributed Constraint Optimization Problem), is proposed to find the best recommendation considering the preferences of the users.

The system works with two types of agents: a user agent (UA), who works on behalf of a user and knows his preferences, and a recommender agent (RA), who works on behalf of suppliers of travel services. An optimization function is proposed to handle the agents' interactions and find the best recommendation.

e-Tourism [23] is a system that plans tourist tours for groups of people. The system considers different aspects, like a group tastes, its demographic classification and places previously visited. A taxonomy-driven recommendation tool called GRSK (Generalist Recommender System Kernel), provides individual recommendations using three techniques: demographic, content-based and preference-based filtering. For each technique group preferences are computed using aggregation, intersection and incremental intersection methods and a list of recommended items is filtered.

Pocket RestaurantFinder [34] is a system that suggests restaurants to groups of people who want to dine together. The system was designed for contexts like conferences, where an occasional group of attendees decides upon a restaurant to visit.

Each user fills a profile with preferences about restaurants, like the price range or the type of cuisine they like (or don't like). Once the group composition is known, the system estimates a user's individual preference for each restaurant and averages those values to build a group preference and produce a list of recommendations.

2.2.3 Group recommendation for TV programs

FIT (Family Interactive TV System) [25] is a recommender system that aims to filter TV programs considering the preferences of the viewers.

The only input required by the system is a stereotype user representation (i.e., a class of viewers that would suit the user, like *women*, *businessmen*, *students*, etc.), along with the user preferred watching time. The system automatically updates a profile, by collecting implicit feedbacks from the watching habits of the user.

When someone starts watching TV, the system looks at the probability of each family member to watch TV in that time slot and predicts who there might be watching the TV. Programs are recommended through an algorithm that combines such probabilities and users' preferences.

The system proposed in [54] recommends TV programs to a family.

To protect the privacy of each user and avoid the sharing of information, the system observes the habits of a user and adds contextual information about what is monitored. By observing indicators like the amount of time a TV program has been watched, a user's preferences are exploited and a profile is built.

To estimate the interests of the users in different aspects, the system trains on each family history three Support Vector Machine (SVM) models for program name, genre and viewing history. After the models are trained, recommendation is performed with a Case-Based Reasoning (CBR) technique.

TV4M [56] is a TV programs recommender system for multiple viewers.

To identify who is watching TV, the system provides a login feature. To build a group profile that satisfies most of its members, all the current viewers' profiles are merged, by doing a total distance minimization of the features available (e.g., genre, actor, etc.). According to the built profile, programs are recommended to the group.

2.3 Systems that consider random groups who share an environment

A random group is formed by people who share an environment without a specific purpose. Its nature is *heterogeneous* and its members might not share interests.

Group recommender systems that work with random groups calculate the list of predicted items frequently, as people might join or leave the environment. This section will describe group recommender systems that work with random groups. Two main recommendation domains are related to this type of systems:

- multimedia items (e.g., music) broadcast in a shared environment;
- information items (e.g., news or web pages).

2.3.1 Group recommendation for broadcast multimedia items

Adaptive Radio [13] is a system that broadcasts songs to a group of people who share an environment. The approach tries to improve satisfaction of the users by focusing on *negative preferences*, i.e., it keeps track of which songs a user does not like and avoids playing them. Moreover, the songs similar to the ones rejected by a user are reject too (the system considers two songs similar if they belong to the same album). The highest rated between the remaining songs is automatically played.

In-Vehicle Multimedia Recommender [57] is a system that aims to select multimedia items for a group of people traveling together.

The system aggregates the profiles of the passengers and merges them using a notion of *distance* between the profiles. Once the profiles are merged, a content-based recommender system is used to compare multimedia items and group preferences.

Flytrap [17] is a group recommender system that selects music to be played in a public room. Since people in a room (i.e., the group members) change frequently, the system was designed to predict the song to play considering the preferences of the users present in the room at the moment of the song selection.

A 'virtual DJ' agent is used to automatically decide the song to play. To build a model of the preferences of each user the agent analyzes the MP3 files played by a user in his/her computer and considers the information available about the music (like similar genres, artists, etc.). The song is selected through a voting system in which an agent represents each user in the room and rates the candidate tracks.

MusicFX [35] is a system that recommends music to members of a fitness center.

Since the group structure (i.e., the people in the room) varies continuously, the system gives the users working out in the fitness center the possibility to login. To let users express their preferences about a particular genre, the system has a database of music genres. The music to play is selected considering the preferences of each user in a summation formula.

2.3.2 Group recommendation for information items

Let's Browse [32] is a system that recommends pages to people browsing the web together. Since the group is random (a user might join or leave the group at any time), the system uses an electronic badge to detect the presence of a user.

The system builds a user profile analyzing the words present in his/her homepage. The group is modeled by a linear combination of the individual profiles and the system analyzes the words that occur in the pages browsed by the group.

The system recommends pages that contain keywords present in the user profile.

GAIN (Group Adapted Interaction for News) [46, 11] is a system that selects background information to display in a public shared environment.

The authors assumed that the group of users may be totally unknown, partially or completely known. The group is modeled by splitting it in two subgroups: the *known subgroup* (i.e., people that are certainly near the display for a period of time) and the *unknown subgroup* (i.e., people not recognized by the system). Recommendations are predicted using a statistical dataset built from the group modeling.

3 Group recommendation with automatic group identification

As shown in Table 1, two group recommender systems automatically detect groups of users. Such an approach is interesting for various reasons: (I) people change their mind frequently, so a user membership in a group might not be long-term, or (II) technological constraints might allow the system to handle only a certain number of groups (or a maximum number of members per group). Group recommender systems that automatically detect groups were developed for the following domains:

- identification of Communities of Interests (groups of similar and previously unrelated people);
- movies recommendation in case of limited bandwidth;

3.1 Group recommendation with Communities of Interest identification

The approach proposed in [10] aims to automatically discover Communities of Interest (CoI) (i.e., a group of individuals who share and exchange ideas about a given interest) and produce recommendations for them.

CoI are identified exploiting the preferences expressed by users in personal ontology-based profiles. Each profile measures the interest of a user in concepts of the ontology. The interest expressed by users is used to cluster the concepts.

User profiles are then split into subsets of interests, to link the preferences of each user with a specific cluster of concepts. Hence it is possible to define relations among users at different levels, obtaining a multilayered interest network that allows to find multiple CoI. Recommendations are built using a content-based CF approach.

3.2 *Group recommendation with automatic identification of users' communities in case of bandwidth limitations*

None of the approaches described takes into account the fact that it might be necessary to identify groups of people with similar interests because of technological constraints, like bandwidth limitations.

For example, in multiple access systems with limited transmission capacity like Mobile IPTV or Satellite Systems, it might not be possible to create personalized program schedules for each user. In such cases, the problem relies in identifying groups of related users to fulfill the constraints.

Here we present an approach proposed in [8] to generate group recommendations, able to detect intrinsic communities of users whose preferences are similar. The algorithm takes as input a matrix that associates a set of *users* to a set of *items* through a *rating*. This matrix will be called the *ratings matrix*. Based on ratings expressed by each user in the ratings matrix, the algorithm evaluates the level of similarity between users and generates a network that contains the similarities. A modularity-based Community Detection algorithm proposed by [7] will be run on the network, to find partitions of users in communities. For each community, ratings for all the items will be calculated.

Since the Community Detection algorithm is able to produce a dendrogram, i.e., a tree that contains hierarchical partitions of the users in communities of increasing granularity, experiments were conducted in order to evaluate the quality of the recommendation for the different partitions. Results show that the quality of group recommendations increases linearly with the number of communities created.

The scientific contribution of the recommendation algorithm is the capability to automatically detect intrinsic communities of users who share similar preferences, making it possible for a content provider to explore the trade off between the level of personalization of the recommendation and the number of channels.

3.2.1 Group recommendation with automatic identification of users communities

The group recommendation algorithm works in four steps:

Users similarity evaluation

In order to create communities of users, the algorithm takes as input a *ratings matrix* and evaluates through a standard metric (cosine similarity) how similar the preferences of two users are. The result is a weighted network where nodes represent users and a weighted edge represents the similarity value of the users it connects.

Communities detection

To identify intrinsic communities of users, a Community Detection algorithm proposed in [7] is applied to the users similarity network and partitions of different granularities are generated.

Ratings prediction for items rated by enough users of a group

A group's ratings are evaluated by calculating, for each item, the mean of the ratings expressed by the users of the group. In order to predict meaningful ratings, the algorithm calculates a rating only if an item was evaluated by a minimum percentage of users in the group. With this step it is not possible to predict a rating for each item, so another step has been created to predict the remaining ratings.

Ratings prediction for the remaining items

For some of the items, ratings could not be calculated by the previous step. In order to estimate such ratings, similarity between items is evaluated, and the rating of an item is predicted considering the items most similar to it.

The four steps that constitute the algorithm will now be described in detail.

Step 1. Users similarity evaluation

Here it is described how a ratings matrix can be used to evaluate similarity between users. Let v_i be the vector of the ratings expressed by a user i for the items and v_j be the vector of the ratings expressed by a user j for the items. The similarity s_{ij} between users i and j can be measured by the cosine similarity between the vectors:

$$s_{ij} = cos(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\| \times \|v_j\|}$$

Similarities can be represented in a network, the *users similarity network*, that links each couple of associated users with a weighted edge.

As highlighted by [24], in networks like the one built, edges have intrinsic weights and no information is given about the real associations between the nodes. Edges are usually affected by noise, which leads to ambiguities in the communities detection. Moreover, the weights of the edges in the network are calculated considering the ratings and it is well known that people have different rating tendencies: some users tend to express their opinion using just the end of the scales, expressing if they loved or hated an item. To eliminate noise from the network and reduce its complexity by removing weak edges, a parameter called *noise* was set in the algorithm. The parameter indicates the weight that will be subtracted by every edge.

Step 2. Communities Detection

This step of the algorithm has the goal to find intrinsic communities of users, accepting as input the weighted users similarity network that was built in the previous step. Another requirement is to produce the intrinsic users communities in a hierarchical structure, in order to deeper understand and exploit its inner partition. Out of all the existing classes of clustering algorithms, complex network analysis [21] was identified as the only class of algorithms fulfilling the requirements. In 2004 an optimization function has been introduced, the modularity [41], that measures for a generic partition of the set of nodes in the network, the number of internal (in each partition) edges respect to the random case. The optimization of this function gives, without a previous assessment of the number and size of the partitions [21], the natural community structure of the network. Moreover it is not necessary to embed the network in a metric space like in the k-means algorithm. A notion of distance or link weight can be introduced but in a pure topological fashion [42].

Recently a very efficient algorithm has been proposed, based on the optimization of the weighted modularity, that is able to easily handle networks with millions of nodes, generating also a dendrogram; a community structure at various network resolutions [7]. Since the algorithm had all the characteristics needed, it was chosen to create the groups of users used by the group recommendation algorithm.

Step 3. Ratings prediction for items rated by enough users of a group

To express a group's preference for an item, the algorithm calculates its rating, considering the ratings expressed by the users of the community for that item.

An average is a single value that is meant to typify a list of values. The most common method to calculate such a value is the arithmetic mean, which also seems an effective way to put together all the ratings expressed by the users in a group. So, for each item i, its rating r_i is expressed as:

$$r_i = \frac{1}{n}\sum_{u=0}^{n} r_u$$

where n is the number of users of the group who expressed a rating for item i and r_u is the rating expressed by each user for that item. In order to calculate meaningful ratings for a group, a rating r_i is considered only if a minimum part of the group has rated the item. This is done through a parameter, called *co-ratings* which expresses the minimum percentage of users who have to rate an item in order to calculate the rating for the group.

Step 4. Ratings prediction for the remaining items

For some of the items, ratings could not be calculated by the previous step. In order to estimate such ratings, a network that contains similarities between items was built. Like the users similarity network presented in 3.2.1, the network is built through the ratings matrix, considering the ratings expressed for each item. Let w_i be the vector of the ratings expressed by all the users for item i and w_j be the vector of the ratings expressed by all the users for item j. The similarity t_{ij} between item i and item j is measured with the cosine similarity and the similarities are represented in a network called *items similarity network*, from which noise was removed through the *noise* parameter presented in 3.2.1.

For each item not rated by the group, a list is produced with its nearest neighbors, i.e., the most similar items already rated by the group, considering the similarities available in the *items similarity network*. Out of this list, the *top* items are selected. Parameter *top* indicates how many similarities the algorithm considers to predict the ratings. An example of how the *top* similar items are selected is shown in Table 2. The algorithm needs to predict a rating for Item 1. The most similar items are shown in the list. For each similar item j, the table indicates the similarity with Item 1 (column t_{1j}) and the rating expressed by the group (column r_j). In the example, the *top* parameter is set to 3 and items with similarity 0.95, 0.88 and 0.71 are selected.

Item j	t_{1j}	r_j
Item 2	0.95	3.5
Item 3	0.95	4.2
Item 4	0.88	2.8
Item 5	0.71	2.6
Item 6	0.71	3.9
Item 7	0.71	4.3
Item 8	0.63	1.2
Item 9	0.55	3.2

Table 2 Top similar items of an unrated item

it is now possible to predict the rating of an unrated item by considering both the rating and the similarity of its *top* similar items:

$$\bar{r}_i = \frac{\sum_{j=0}^{n} r_j \cdot t_{ij}}{\sum_{j=0}^{n} t_{ij}}$$

where n is the number of items selected in the list. Given the example in Table 2, $\bar{r}_1 = 3.55$.

To make meaningful predictions, an evaluation of how "reliable" the predictions are is needed. This is done by calculating the mean of the *top* similarities and by setting a *trust* parameter. The parameter indicates the minimum value the mean of the similarities has to get, in order to be considered reliable and consider the predicted rating. The mean of the similarities in the previous example is 0.85 so, to consider $\bar{r}_1$, the *trust* parameter has to be lower than 0.85.

3.2.2 Algorithm Experimentation

To evaluate the quality of the recommendations, the algorithm was tested using MovieLens[2], a dataset widely used to evaluate CF algorithms. A framework that extracts a subset of ratings from the dataset, predicts group recommendations through the presented algorithm and measures the quality of the predictions in terms of RMSE was built. Details of the algorithm experimentation will now be described.

Experimental methodology and setup

The experimentation was made with the MovieLens dataset, which is composed of 1 million ratings, expressed by 6040 users for 3900 movies. To evaluate the quality of the ratings predicted by the algorithm, around 10% of the ratings was extracted as a probe test set and the rest of the dataset was used as a training set for the algorithm.

The group recommendation algorithm was run with the training set and, for each partition of the users in communities, ratings were predicted. The quality of the predicted ratings was measured through the Root Mean Squared Error (RMSE). The metric compares the probe test set with the ratings predicted: each rating r_i expressed by a user u for an item i is compared with the rating $\bar{r}_i$ predicted for the item i for the group in which user u is. The formula is shown below:

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(r_i - \bar{r}_i)^2}{n}}$$

where n is the number of ratings available in the test set. To evaluate the performances of the algorithm, they were compared with the results obtained considering a single group with all the users (predictions are calculated considering all the preferences expressed for an item), and the results obtained using a classic CF algorithm proposed in [15], where recommendations are produced for each user.

Experimental results

To evaluate the algorithm's performances the quality of the recommendations was studied, considering different values of each parameter. The only value that could

[2] http://www.grouplens.org/

not be changed was *noise*, because if more than 0.1 was subtracted to the edges of the *users similarities network*, the network would become disconnected.

The first experiment conducted evaluated the quality of the recommendations for different values of the *co-ratings* parameter, i.e., the minimum percentage of users who have to rate an item, in order to calculate the rating for the group. Parameter *top* was set to 2 and parameter *trust* was set to 0.0. Fig. 1 shows how RMSE varies with the number of groups, for different values of *co-ratings* (10%, 20% and 25%). It is

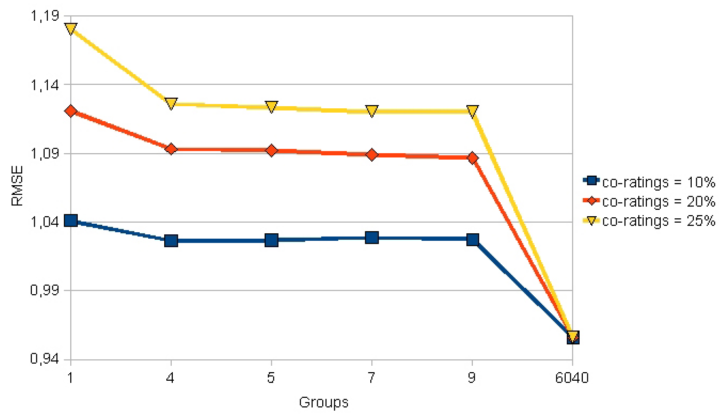

Fig. 1 Algorithm's performances with different co-ratings values

possible to see that as the number of groups grows, the quality of the recommendations improves, since groups get smaller and the algorithm predicts more precise ratings. To conduct the following experiments, the value of *co-ratings* chosen was 20%. The next experiment conducted was to evaluate the quality of recommendations for different values of the *top* parameter, i.e., the number of similarities considered to select the nearest neighbors of an item. Fig. 2 shows how RMSE varies with the number of groups, for different values of *top* (2 and 3). It is worth noting that the quality of the recommendations improves when parameter *top* is set to 3 (i.e., the top 3 similarities are selected from the list), so this was the value set for the next experiment. The last parameter to evaluate is *trust*, i.e., the minimum value the mean of the similarities has to get when the algorithms predicts a rating considering the nearest neighbors of an item. Fig. 3 shows how RMSE varies with the number of groups, for different values of the parameter (0.0, 0.1 and 0.2). In Fig. 3 is shown that the quality of the performances improves for higher values of *trust*, i.e., when the ratings predicted can be considered more "reliable".

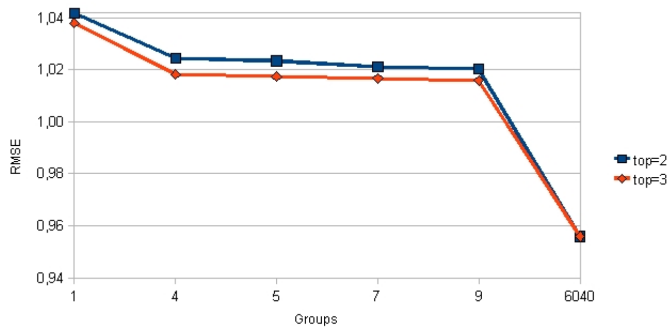

Fig. 2 Algorithm's performances for different values of top

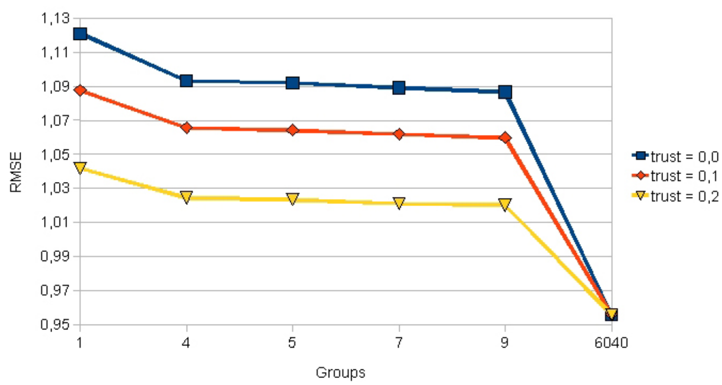

Fig. 3 Algorithm's performances with different trust values

4 Conclusions and discussion

Recommender systems have become important tools that help people making decisions, by adapting to preferences or characteristics of a user and effectively suggesting items that might interest him/her. However, there are contexts in which people operate in groups and in the last years several approaches to produce recommendations for groups of users were developed.

This chapter presented a state-of-the-art survey on group recommendation, focusing on the nature of the group considered by each system. Moreover, a new approach able to adapt to technological constraints (e.g., bandwidth limitations) and produce recommendations for automatically detected groups was presented.

As we can see, nearly all the approaches take for granted the type of group they are aimed to: whether the group is *established*, *occasional* or *random*, its structure is taken "as is". However, there might be contexts in which groups are not available and just two approaches focus on the identification of groups. We believe that the study of algorithms specifically designed for group recommendation, able to model and identify groups, might improve the quality of the recommendation process.

References

1. Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawla, Gautam Das, and Cong Yu. Group recommendation: Semantics and efficiency. *PVLDB*, 2(1):754–765, 2009.
2. Liliana Ardissono, Anna Goy, Giovanna Petrone, and Marino Segnan. A multi-agent infrastructure for developing personalized web-based systems. *ACM Trans. Internet Technol.*, 5(1):47–69, 2005.
3. Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence*, 17(8):687–714, 2003.
4. Claudio Baccigalupo and Enric Plaza. A case-based song scheduler for group customised radio. In Weber and Richter [55], pages 433–448.
5. Jacob P. Baskin and Shriram Krishnamurthi. Preference aggregation in group recommender systems for committee decision-making. In Bergman et al. [6], pages 337–340.
6. Lawrence D. Bergman, Alexander Tuzhilin, Robin D. Burke, Alexander Felfernig, and Lars Schmidt-Thieme, editors. *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*. ACM, 2009.
7. Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008+, October 2008.
8. Ludovico Boratto, Salvatore Carta, Alessandro Chessa, Maurizio Agelli, and M. Laura Clemente. Group recommendation with automatic identification of users communities. In *Web Intelligence/IAT Workshops*, pages 547–550. IEEE, 2009.
9. Peter Briggs and Barry Smyth. Modeling trust in collaborative web search. In *AICS*, Coleraine, NI, 2005.
10. Ivn Cantador, Pablo Castells, and Escuela Politcnica Superior. Extracting multilayered semantic communities of interest from ontology-based user profiles: Application to group modelling and hybrid recommendations. In *In: Computers in Human Behavior, special issue on Advances of Knowledge Management and the Semantic*. Elsevier. In press, 2008.
11. Berardina De Carolis and Sebastiano Pizzutilo. Providing relevant background information in smart environments. In Noia and Buccafurri [43], pages 360–371.
12. S. Carta, A. Alimonda, M.L. Clemente, and M. Agelli. Glue: Improving tag-based contents retrieval exploiting implicit user feedback. In Eduard Hoenkamp, Martine de Cock, and Veronique Hoste, editors, *Proceedings Of The 8th Dutch-Belgian Information Retrieval Workshop (DIR 2008)*, pages 29–35, 2008.
13. Dennis L. Chao, Justin Balthrop, and Stephanie Forrest. Adaptive radio: achieving consensus using negative preferences. In Mark Pendergast, Kjeld Schmidt, Gloria Mark, and Mark Ackerman, editors, *GROUP*, pages 120–123. ACM, 2005.
14. Yen-Liang Chen, Li-Chen Cheng, and Ching-Nan Chuang. A group recommendation system with consideration of interactions among group members. *Expert Syst. Appl.*, 34(3):2082–2090, 2008.
15. Maria Laura Clemente. Experimental results on item-based algorithms for independent domain collaborative filtering. In *AXMEDIS '08: Proceedings of the 2008 International Conference on Automated solutions for Cross Media Content and Multi-channel Distribution*, pages 87–92, Washington, DC, USA, 2008. IEEE Computer Society.
16. Maurice Coyle and Barry Smyth. Explaining search results. In Kaelbling and Saffiotti [30], pages 1553–1555.

17. Andrew Crossen, Jay Budzik, and Kristian J. Hammond. Flytrap: intelligent group music recommendation. In *IUI*, pages 184–185, 2002.
18. Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Group recommending: A methodological approach based on bayesian networks. In *ICDE Workshops*, pages 835–844. IEEE Computer Society, 2007.
19. Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Managing uncertainty in group recommending processes. *User Model. User-Adapt. Interact.*, 19(3):207–242, 2009.
20. Paul H. Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *UIST*, pages 219–226, 2001.
21. Santo Fortunato and Claudio Castellano. Community structure in graphs. *Springer's Encyclopedia of Complexity and System Science*, Dec 2007.
22. Jill Freyne and Barry Smyth. B.: Cooperating search communities. In *Proc. of 4th International Conference on Adaptive Hypermedia and Adaptive WebBased Systems (AH'2006). Lecture Notes in Computer Science*, pages 101–111. Springer Verlag, 2006.
23. Inma Garcia, Laura Sebastia, Eva Onaindia, and Cesar Guzman. A group recommender system for tourist activities. In Noia and Buccafurri [43], pages 26–37.
24. D. Gfeller, J. C. Chappelier, and De Los. Finding instabilities in the community structure of complex networks. *Physical Review E*, 72(5 Pt 2):056135+, November 2005.
25. Dina Goren-Bar and Oded Glinansky. Fit-recommend ing tv programs to family members. *Computers & Graphics*, 28(2):149–156, 2004.
26. Anthony Jameson. More than the sum of its members: Challenges for group recommender systems. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 48–54, Gallipoli, Italy, 2004. Available from http://dfki.de/∼jameson/abs/Jameson04AVI.html.
27. Anthony Jameson, Stephan Baldes, and Thomas Kleinbauer. Enhancing mutual awareness in group recommender systems. In Bamshad Mobasher and Sarabjot S. Anand, editors, *Proceedings of the IJCAI 2003 Workshop on Intelligent Techniques for Web Personalization*. AAAI, Menlo Park, CA, 2003. Available from http://dfki.de/∼jameson/abs/JamesonBK03ITWP.html.
28. Anthony Jameson, Stephan Baldes, and Thomas Kleinbauer. Two methods for enhancing mutual awareness in a group recommender system. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, Gallipoli, Italy, 2004. In press.
29. Anthony Jameson and Barry Smyth. Recommendation to groups. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 596–627. Springer, 2007.
30. Leslie Pack Kaelbling and Alessandro Saffiotti, editors. *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*. Professional Book Center, 2005.
31. Jae Kyeong Kim, Hyea Kyeong Kim, Hee Young Oh, and Young U. Ryu. A group recommendation system for online communities. *International Journal of Information Management*, In Press, Corrected Proof:–, 2009.
32. Henry Lieberman, Neil W. Van Dyke, and Adriana Santarosa Vivacqua. Let's browse: A collaborative web browsing agent. In *IUI*, pages 65–68, 1999.
33. Fabiana Lorenzi, Fernando Santos, Paulo R. Ferreira, Jr., and Ana L. Bazzan. Optimizing preferences within groups: A case study on travel recommendation. In *SBIA '08: Proceedings of the 19th Brazilian Symposium on Artificial Intelligence*, pages 103–112, Berlin, Heidelberg, 2008. Springer-Verlag.
34. Joseph F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, Minneapolis, 2002.
35. Joseph F. McCarthy and Theodore D. Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *CSCW*, page 348, 2000.
36. Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Case-based group recommendation: Compromising for success. In Weber and Richter [55], pages 299–313.

37. Kevin McCarthy, Lorraine McGinty, Barry Smyth, and Maria Salamó. The needs of the many: A case-based group recommender system. In Thomas Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *ECCBR*, volume 4106 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2006.
38. Kevin Mccarthy, Lorraine Mcginty, Barry Smyth, and Maria Salamo. Social interaction in the cats group recommender. In Peter Brusilovsky, Jon Dron, and Jaakko Kurhila, editors, *Workshop on the Social Navigation and Community-Based Adaptation Technologies at the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, June 2006.
39. Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. Cats: A synchronous approach to collaborative group recommendation. In Geoff Sutcliffe and Randy Goebel, editors, *FLAIRS Conference*, pages 86–91. AAAI Press, 2006.
40. Kevin McCarthy, Maria Salamó, Lorcan Coyle, Lorraine McGinty, Barry Smyth, and Paddy Nixon. Group recommender systems: a critiquing based approach. In Cécile Paris and Candace L. Sidner, editors, *IUI*, pages 267–269. ACM, 2006.
41. M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2 Pt 2), February 2004.
42. M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70(5):056131, Nov 2004.
43. Tommaso Di Noia and Francesco Buccafurri, editors. *E-Commerce and Web Technologies, 10th International Conference, EC-Web 2009, Linz, Austria, September 1-4, 2009. Proceedings*, volume 5692 of *Lecture Notes in Computer Science*. Springer, 2009.
44. Mark O'Connor, Dan Cosley, Joseph A. Konstan, and John Riedl. Polylens: a recommender system for groups of users. In *ECSCW'01: Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work*, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers.
45. Kenton O'Hara, Matthew Lipson, Marcel Jansen, Axel Unger, Huw Jeffries, and Peter Macer. Jukola: democratic music choice in a public space. In *DIS '04: Proceedings of the 5th conference on Designing interactive systems*, pages 145–154, New York, NY, USA, 2004. ACM.
46. Sebastiano Pizzutilo, Berardina De Carolis, Giovanni Cozzolongo, and Francesco Ambruoso. Group modeling in a public space: methods, techniques, experiences. In *AIC'05: Proceedings of the 5th WSEAS International Conference on Applied Informatics and Communications*, pages 175–180, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).
47. Juan A. Recio-García, Guillermo Jiménez-Díaz, Antonio A. Sánchez-Ruiz-Granados, and Belén Díaz-Agudo. Personality aware recommendations to groups. In Bergman et al. [6], pages 325–328.
48. Taly Sharon, Henry Lieberman, and Ted Selker. A zero-input interface for leveraging group experience in web browsing. In *IUI*, pages 290–292. ACM, 2003.
49. B. Smyth, J. Freyne, M. Coyle, P. Briggs, and E. Balfe. I-SPY: Anonymous, Community-Based Personalization by Collaborative Web Search. In *Proceedings of the 23rd SGAI International Conference on Innovative Techniques*, pages 367–380. Springer, 2003. Cambridge, UK.
50. Barry Smyth and Evelyn Balfe. Anonymous personalization in collaborative web search. *Inf. Retr.*, 9(2):165–190, 2006.
51. Barry Smyth, Evelyn Balfe, Oisín Boydell, Keith Bradley, Peter Briggs, Maurice Coyle, and Jill Freyne. A live-user evaluation of collaborative web search. In Kaelbling and Saffiotti [30], pages 1419–1424.
52. Barry Smyth, Evelyn Balfe, Peter Briggs, Maurice Coyle, and Jill Freyne. Collaborative web search. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 1417–1419. Morgan Kaufmann, 2003.
53. David Sprague, Fuqu Wu, and Melanie Tory. Music selection using the partyvote democratic jukebox. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 433–436, New York, NY, USA, 2008. ACM.

54. Elena Vildjiounaite, Vesa Kyllönen, Tero Hannula, and Petteri Alahuhta. Unobtrusive dynamic modelling of tv programme preferences in a finnish household. *Multimedia Syst.*, 15(3):143–157, 2009.
55. Rosina Weber and Michael M. Richter, editors. *Case-Based Reasoning Research and Development, 7th International Conference on Case-Based Reasoning, ICCBR 2007, Belfast, Northern Ireland, UK, August 13-16, 2007, Proceedings*, volume 4626 of *Lecture Notes in Computer Science*. Springer, 2007.
56. Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. Tv program recommendation for multiple viewers based on user profile merging. *User Model. User-Adapt. Interact.*, 16(1):63–82, 2006.
57. Yu Zhiwen, Zhou Xingshe, and Zhang Daqing. An adaptive in-vehicle multimedia recommender for group users. In *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, volume 5, pages 2800–2804 Vol. 5, 2005.